

DEVICE AND METHOD FOR MODULATION, DEVICE AND METHOD FOR DEMODULATION AND PROVIDING MEDIUM

Publication number: JP11346154 (A)

Publication date: 1999-12-14

Inventor(s): NAKAGAWA TOSHIYUKI; NIIFUKU YOSHIHIDE; NARAHARA TATSUYA; NAKAMURA KOSUKE; KEESU A SUHAUHAAMERU IMINKU; G J VAN DEN ENDEN; J A H M KARLMANN +

Applicant(s): SONY CORP; PHILIPS ELECTRONICS NV +

Classification:

- **international:** G06F7/00; G11B20/14; G11B21/21; G11B5/60; H03M5/14; H03M7/02; H03M7/14; H03M7/40; G06F7/00; G11B; G11B20/14; G11B21/21; G11B5/60; H03M; H03M5/00; H03M7/02; H03M7/14; H03M7/40; (IPC1-7): H03M7/14

- **European:** G11B20/14A2B; H03M5/14B

Application number: JP19980150280 19980529

Priority number(s): JP19980150280 19980529

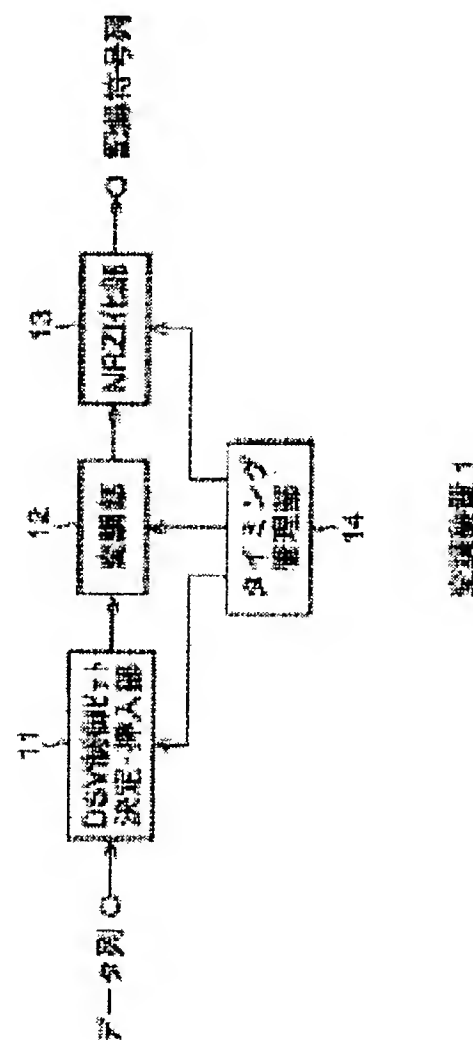
Also published as:

JP3985173 (B2)
US6496541 (B1)
ZA200000371 (A)
US2007063881 (A1)
US7466246 (B2)

more >>

Abstract of JP 11346154 (A)

PROBLEM TO BE SOLVED: To record and to reproduce at high linear density. **SOLUTION:** A DSV control bit decision/insertion part 11 inserts a DSV control bit for DSV control in an inputted data sequence and outputs it to a modulation part 12. The modulation part 12 converts the data of two bit basic data length into a variable length code of three bit basic code length and outputs them to a non return to zero inverted(NRZI) part 13 in accordance with a conversion table.; A conversion table which the modulation part 12 possesses has a replacement code for restricting continuation of a minimum run below the specified number of times and a replacement code for protecting a run length restriction, and moreover, a conversion rule which has a remainder after the number of '1' in an element of the data sequence is divided, by 2, and the remainder after the number of '1' in an element of a code word sequence is divided by 2 agree with each other as 1 or 0.



Data supplied from the **espacenet** database — Worldwide

(51) Int.Cl.⁹
H 0 3 M 7 / 1 4

識別記号

F I
H 0 3 M 7 / 1 4

B

審査請求 未請求 請求項の数31 O L (全 24 頁)

(21) 出願番号 特願平10－150280

(22) 出願日 平成10年(1998) 5 月29日

(71) 出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号
598070935

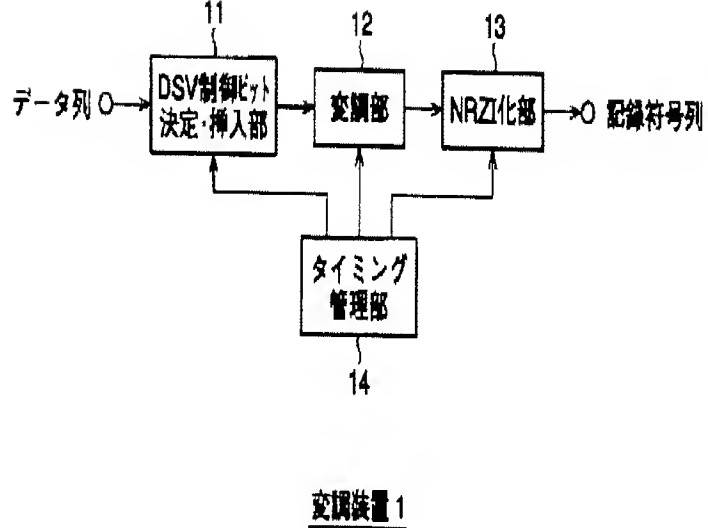
(71) 出願人
ライリツクス・エレクトロニクス・エヌ・
アイ
Koninklijke Philips
Electronics NV.
オランダ アイントホーフェン市 ビイ・
エイ5621 フローネンボウセ通り 1

(74) 代理人 弁理士 稲本 義雄

(54) 【発明の名称】 変調装置および方法、復調装置および方法、並びに提供媒体

最終頁に続く

(57) 【要約】
【課題】 高線密度での記録再生ができるようにする。
【解決手段】 DSV制御ビット決定・挿入部11は、入力されたデータ列にDSV制御のためのDSV制御ビットを挿入し、変調部12に出力する。変調部12は、変換テーブルに従って、基本データ長が2ビットのデータを、基本符号長が3ビットの可変長符号に変換して、NRZ1化部13に出力する。変調部12が有する変換テーブルは、最小ランの連続を所定の回数以下に制限する置き換えコード、さらに、データ列の要素内の「1」の個数を2で割ったときの余りと、符号語列の要素内の「1」の個数を2で割ったときの余りが、どちらも1あるいは0で一致するような変換規則を有する。



【特許請求の範囲】

【請求項 1】 基本データ長が m ビットのデータを、基本符号長が n ビットの可変長符号 (d, k ; m, n ; r) に変換する変調装置において、入力されたデータを、変換テーブルに従って、符号に変換する変換手段を備え、

前記変換テーブルの変換コードは、

$d = 1, k = 7, m = 2, n = 3$ の基礎コードと、

データ列の要素内の「1」の個数を 2 で割った時の余りと、変換される符号語列の要素内の「1」の個数を 2 で

割った時の余りが、どちらも 1 あるいは 0 で一致するよ

うな変換規則と、

最小ラウンドの連続を所定の回数以下に制限する第 1 の置き換えコードと、

ラン長制限を守るための第 2 の置き換えコードとを有すること

を特徴とする変調装置。

【請求項 2】 拘束長 i = 1 における前記基礎コードを構成するデータ列と符号語列の対の数は、 $2^m = 2^2 = 4$ よりも少ないことを特徴とする請求項 1 に記載の変調装置。

【請求項 3】 前記変換テーブルの基礎コードは、可変長構造を有することを特徴とする請求項 1 に記載の変調装置。

【請求項 4】 前記変換テーブルの基礎コードは、不確定符号を含み、前記不確定符号は、直前または直後の符号語が「1」のとき「0」となり、「0」のとき「1」となる記号を*とすると、 0000 または 1101 となる符号「*0*」を含むことを特徴とする請求項 1 に記載の変調装置。

【請求項 5】 前記変換テーブルの変換コードは、直後の符号語列、または、直後に続くデータ列を参照して決定するコードを含むことを特徴とする請求項 1 に記載の変調装置。

【請求項 6】 前記参照する直後の符号語列は、特定の種類の符号語列とすることを特徴とする請求項 5 に記載の変調装置。

【請求項 7】 前記直後の符号語列、または、直後に続くデータ列を参照して決定する変換コードは、前記第 1 または第 2 の置き換えコードであることを特徴とする請求項 5 に記載の変調装置。

【請求項 8】 拘束長 i が 1 である場合における前記基礎コードを構成するデータ列と符号語列の対の数は、 $2^m = 2^2 = 4$ に等しいことを特徴とする請求項 1 に記載の変調装置。

【請求項 9】 拘束長 i が 2 以上の場合の前記変換コードは、すべて前記第 1 または第 2 の置き換えコードであることを特徴とする請求項 1 に記載の変調装置。

【請求項 10】 前記置き換えコードは、拘束長 i が 2 の場合の変換コードは、最小ラウンド = 1 を守るコードであることを特徴とする請求項 1 に記載の変調装置。

【請求項 11】 前記変換テーブルの変換コードは、直前の符号語列を参照して決定するコードを含むことを特徴とする請求項 10 に記載の変調装置。

【請求項 12】 前記変換テーブルの変換コードとして存在しないユニークなパターンを含む同期信号を、前記符号語列の任意の位置に挿入する挿入手段をさらに備えることを特徴とする請求項 1 に記載の変調装置。

【請求項 13】 前記同期信号に用いられるユニークなパターンは、最大ラウンドを破るパターンであることを特徴とする請求項 12 に記載の変調装置。

【請求項 14】 前記同期信号に用いられるユニークなパターンは、最小ラウンドを守るパターンであることを特徴とする請求項 12 に記載の変調装置。

【請求項 15】 前記同期信号は、先頭の 1 符号語が、直前までのデータを変換した符号語との接続ビットであり、2 番目の符号語が、最小ラウンドを守るためのビットであり、3 番目の符号語から、前記同期信号としてユニークなパターンを構成することを特徴とする請求項 12 に記載の変調装置。

【請求項 16】 前記同期信号の大きさは、少なくとも 12 符号語であることを特徴とする請求項 12 に記載の変調装置。

【請求項 17】 前記同期信号は、その大きさが、21 符号語以上るとき、最大ラウンド = 8 のパターンを少なくとも 2 個含むことを特徴とする請求項 12 に記載の変調装置。

【請求項 18】 前記変換テーブルの変換コードは、符号を任意の位置において終端させるための終端コードをさらに有することを特徴とする請求項 12 に記載の変調装置。

【請求項 19】 前記終端コードは、前記データ列と符号語列の対の数が、 $2^m = 2^2 = 4$ よりも少ない前記拘束長 i の前記基礎コードに対応して規定され、かつ、データ列の要素内の「1」の個数を 2 で割った時の余りと、変換される符号語列の要素内の「1」の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で一致するよ

うな変換規則を有することを特徴とする請求項 18 に記載の変調装置。

【請求項 20】 前記終端コードを識別するために、前記同期信号パターンにおいて接続ビットとされた、先頭の 1 符号語ビットを、前記終端コードを用いたときには「1」とし、そうでないときは「0」とすることを特徴とする請求項 18 に記載の変調装置。

【請求項 21】 前記同期信号は、その先頭の 3 ビット、及び、最後の 3 ビットが、データ列及び接続用の混合した接続ビットとされ、その間が、前記ユニークなパターンとされることが特徴とする請求項 12 に記載の変調装置。

【請求項 22】 前記同期信号の先頭の 3 ビットは、変換前のデータ語において、m ビット単位で見ても、先頭の

50

1ビット目をデータ語に対応する値とし、次の2ビット目を前記同期信号を規定するために「1」とし、前記同期信号の最後の3ビットは、変換前のデータ語において、mビット単位で見て、先頭の1ビット目は同期信号を規定するために「0」とし、次の2ビット目をデータ語に対応する値とすることを特徴とする請求項12に記載の変調装置。

【請求項23】 入力されたデータのDSVを制御して、前記変換手段に供給するDSV制御手段をさらに備えることを特徴とする請求項1に記載の変調装置。

10 【請求項24】 前記変換手段は、前記最小ラウンドの連続を制限する前記第1の置き換えコードを検出する第1の検出手段と、前記ラウンド長制限を守る前記第2の置き換えコードを検出する第2の検出手段とを備えることを特徴とする請求項1に記載の変調装置。

20 【請求項25】 基本データ長がmビットのデータを、基本符号長がnビットの可変長符号(d, k ; m, n ; r)に変換する変調装置において、入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含み、前記変換テーブルの変換コードは、

d = 1、k = 7、m = 2、n = 3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラウンドの連続を所定の回数以下に制限する第1の置き換えコードと、ラウンド長制限を守るための第2の置き換えコードとを有することを特徴とする変調方法。

30 【請求項26】 基本データ長がmビットのデータを、基本符号長がnビットの可変長符号(d, k ; m, n ; r)に変換する変調装置に、入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含む処理を実行させるプログラムを提供する提供媒体であって、前記変換テーブルの変換コードは、

40 d = 1、k = 7、m = 2、n = 3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラウンドの連続を所定の回数以下に制限する第1の置き換えコードと、ラウンド長制限を守るための第2の置き換えコードとを有することを特徴とする提供媒体。

【請求項27】 基本符号長がnビットの可変長符号(d, k ; m, n ; r)を、基本データ長がmビットのデータに変換する復調装置において、

入力された符号を、変換テーブルに従って、データに変換する変換手段を備え、前記変換テーブルの変換コードは、

d = 1、k = 7、m = 2、n = 3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラウンドの連続を所定の回数以下に制限する第1の置き換えコードと、ラウンド長制限を守るための第2の置き換えコードとを有することを特徴とする復調装置。

10 【請求項28】 所定の間隔で挿入された冗長ビットを除去する除去手段をさらに備えることを特徴とする請求項27に記載の復調装置。

【請求項29】 前記冗長ビットは、DSV制御ビット、または同期信号であることを特徴とする請求項28に記載の復調装置。

20 【請求項30】 基本符号長がnビットの可変長符号(d, k ; m, n ; r)を、基本データ長がmビットのデータに変換する復調装置の復調方法において、入力された符号を、変換テーブルに従って、データに変換する変換ステップを含み、前記変換テーブルの変換コードは、d = 1、k = 7、m = 2、n = 3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラウンドの連続を所定の回数以下に制限する第1の置き換えコードと、ラウンド長制限を守るための第2の置き換えコードとを有することを特徴とする復調方法。

30 【請求項31】 基本符号長がnビットの可変長符号(d, k ; m, n ; r)を、基本データ長がmビットのデータに変換する復調装置に、入力された符号を、変換テーブルに従って、データに変換する変換ステップを含む処理を実行させるプログラムを提供する提供媒体であって、前記変換テーブルの変換コードは、

40 d = 1、k = 7、m = 2、n = 3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラウンドの連続を所定の回数以下に制限する第1の置き換えコードと、ラウンド長制限を守るための第2の置き換えコードとを有することを特徴とする提供媒体。
【発明】の詳細な説明】

【00001】
【発明の属する技術分野】 本発明は、変調装置および方法、復調装置および方法、並びに提供媒体に関し、特に、データを、記録媒体に、高密度に記録または再生する場合に用いて好適な変調装置および方法、復調装置および方法、並びに提供媒体に関する。

【00002】
【従来の技術】 データを所定の伝送路に伝送したり、または例えば磁気ディスク、光ディスク、光磁気ディスク等の記録媒体に記録する際、伝送路や記録媒体に適するように、データの変調が行われる。このような変調方法の1つとして、プロック符号が知られている。このプロック符号は、データ列を $m \times i$ ビットからなる単位（以下データ語という）にプロック化し、このデータ語を適当な符号則に従って、 $n \times i$ ビットからなる符号語に変換するものである。そしてこの符号は、 $i=1$ のときには固定長符号となり、また i が複数個選べるとき、すなわち、 1 乃至 i_{\max} （最大の i ）の範囲の所定の i を選択して変換したときには可変長符号となる。このプロック符号化された符号は可変長符号（ $d, k; m, n; r$ ）と表される。

【00003】 ここで i は拘束長と称され、 i_{\max} は r （最大拘束長）となる。また d は、連続する“1”の間に入る、“0”の最小連続個数、例えば“0”の最小ランを示し、 k は連続する“1”の間に入る、“0”の最大連続個数、例えば“0”の最大ランを示している。

【00004】 ところで上述のようにして得られる可変長符号を、光ディスクや光磁気ディスク等に記録する場合、例えばコンパクトディスク（CD）やミニディスク（MD）では、可変長符号を、“1”を反転とし、“0”を無反転として、NRZI (Non Return to Zero Inverted) 変調し、NRZI 変調された可変長符号（以下、記録波形列とも称する）に基づき記録が行なわれている。また、記録密度のあまり大きくなかった初期のISO規格の光磁気ディスクでは、記録変調されたビット列が、NRZI 変調されず、そのまま記録されていた。

【00005】 記録波形列の最小反転間隔を T_{\min} とし、最大反転間隔を T_{\max} とするとき、線速方向に高密度に記録を行うためには、最小反転間隔 T_{\min} は長い方が、すなわち最小ラン d は大きい方が良く、またクロックの再生の面からは、最大反転間隔 T_{\max} は短い方が、すなわち最大ラン k は小さい方が望ましく、この条件を満足するために、種々の変調方法が提案されている。

【00006】 具体的には、例えば光ディスク、磁気ディスク、又は光磁気ディスク等において、提案あるいは実際に使用されている変調方式として、可変長符号であるRLL（1ー7）（（1, 7; $m, n; r$ ）とも表記される）やRLL（2ー7）（（2, 7; $m, n; r$ ）とも表記される）、そしてISO規格MOに用いられている固定長RLL（1ー7）（（1, 7; $m, n; 1$ ）とも表記

される）などがある。現在開発研究されている、記録密度の高い光ディスクや光磁気ディスク等のディスク装置では、最小ラン $d=1$ のRLL符号（Run Length Limited Code）がよく用いられている。

【00007】 可変長RLL（1ー7）符号の変換テーブルは、例えば以下のようなテーブルである。

＜表1＞

RLL（1, 7; 2, 3; 2）			
データ		符号	
1=1	11	00x	
	10	010	
1=2	01	10x	
	0011	000	00x
	0010	000	010
	0001	100	00x
	0000	100	010

【00008】 ここで変換テーブル内の記号 x は、次に続くチャネルビットが“0”であるときに“1”とされ、また次に続くチャネルビットが“1”であるときに“0”とされる。最大拘束長 r は2である。

【00009】 可変長RLL（1ー7）のパラメータは（1, 7; 2, 3; 2）であり、記録波形列のビット間隔を T とすると、 $(d+1)T$ で表される最小反転間隔 T_{\min} は $2(=1+1)T$ となる。データ列のビット間隔を T_{data} とすると、この $(m/n) \times 2$ で表される最小反転間隔 T_{\min} は $1, 33(=(2/3) \times 2)T_{\text{data}}$ となる。また $(k+1)T$ で表される最大反転間隔 T_{\max} は $8(=7+1)T(=(m/n) \times 8T_{\text{data}}= (2/3) \times 8T_{\text{data}}=5, 33T_{\text{data}})$ である。さらに検出窓幅 T_{wld} は $(m/n) \times T_{\text{data}}$ で表され、その値は $0, 67(=2/3)T_{\text{data}}$ となる。

【0010】 ところで、表1のRLL（1ー7）による変調を行ったチャネルビット列においては、発生頻度としては T_{\min} である $2T$ が一番多く、以下 $3T, 4T$ と続く。 $2T$ や $3T$ のようなエッジ情報が早い周期で多く発生するのは、クロック再生には有利となる場合が多い。

【0011】 ところが、さらに記録線密度を高くしていくと、今度は逆に、最小ランが問題となってくる。すなわち最小ランである $2T$ が連続して発生し続けると、記録波形に歪みが生じやすくなる。なぜなら、 $2T$ の波形出力は、他の波形出力よりも小さく、例えばデフオーカスやタラシェンシャル・チャルト等による影響を受け易いからである。またさらに、高線密度記録の際には、最小ラン $2T$ の連続した記録はノイズ等の外乱の影響も受け易くなり、従ってデータ再生時に誤りが起こり易くなる。この場合におけるデータ再生誤りのパターンのしては、連続する最小ランの先頭と最後がシフトして誤るケースが多く、その結果、発生するビットエラー長が長くなってしまふことになる。

【0012】 ところで、記録媒体へのデータの記録、あ

るいは、データの伝送の際には、記録媒体あるいは伝送路に適した符号化変調が行われるが、これら変調符号に直流成分が含まれていると、例えば、デイク装置のサボの制御におけるトランスキングエラーなどの、各種のエラー信号に変動が生じ易くなったり、あるいはジッターが発生し易くなったりする。従って、変調符号には、直流成分をなるべく含めないようにする方が良い。

【0013】そこで、DSV(Digital Sum Value)を制御することが提案されている。このDSVとは、チャネルビット列をNRZI化し(すなわちレベル符号化し)、そのビット列(データのシンボル)の“1”を「+1」、「0”を「-1」として、符号を加算していったときのその総和を意味する。DSVは符号列の直流成分の目安となる。DSVの絶対値を小さくすること、すなわち、DSV制御を行うことは、符号列の直流成分を抑制することになる。

【0014】上記表1に示した、可変長RLL(1-7)テーブルによる変調符号は、DSV制御が行われている。このような場合のDSV制御は、変調後の符号化列(チャネルビット列)において、所定の間隔でDSV計算を行い、所定のDSV制御ビットを符号化列(チャネルビット列)内に挿入することで、実現する。

【0015】しかしながら、DSV制御ビットは、基本的には冗長ビットである。従って符号変換の効率から考えれば、DSV制御ビットはなるべく少ない方が良い。

【0016】またさらに、挿入されるDSV制御ビットによつて、最小ランドおよび最大ランクは、変化しない方が良い。(d, k)が変化すると、記録再生特性に影響を及ぼしてしまうからである。

【0017】
【発明が解決しようとする課題】以上のように、RLL符号を高線密度にデイクに記録再生する場合、最小ランドの連続したパターンがあると、長いエラーが発生し易いという課題があった。

【0018】また、RLL(1-7)符号のようなRLL符号において、DSV制御を行う場合には、符号語列(チャネルビット列)内に、任意の間隔で、DSV制御ビットを入れる必要があった。DSV制御ビットは冗長であるから、なるべく少ない方が望ましいが、最小ランであるいは最大ランを守るためには、DSV制御ビットが少なくとも2ビット以上必要であり、DSV制御ビットをより短くすることが望まれている。

【0019】本発明は、このような状況に鑑みてなされたものであり、最小ラン $d=1$ であるRLL符号(d, k; m, n) = (1, 7; 2, 3)において、最小ランの連続する回数を制限し、さらに最小ラン及び最大ランを守りながら、効率の良い制御ビットで、DSV制御を行うことができるようにすることを目的とする。

【0020】また本発明は、なるべく単純な構造の変換テーブルを用いて、復調エラー伝搬が増大するのを抑制するようにすることを目的とする。

【0021】
【課題を解決するための手段】請求項1に記載の変調装置は、入力されたデータを、変換テーブルに従って、符号に変換する変換手段を備え、変換テーブルの変換コードは、 $d=1$ 、 $k=7$ 、 $m=2$ 、 $n=3$ の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン d の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0022】請求項25に記載の変調方法は、入力されたデータを、変換テーブルに従って、符号に変換する変換スレッツを含み、変換テーブルの変換コードは、 $d=1$ 、 $k=7$ 、 $m=2$ 、 $n=3$ の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン d の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0023】請求項26に記載の提供媒体は、基本データ長が m ビットのデータを、基本符号長が n ビットの可変長符号(d, k; m, n; r)に変換する変調装置に、入力されたデータを、変換テーブルに従って、符号に変換する変換スレッツを含む処理を実行させるプログラムを提供する提供媒体であつて、変換テーブルの変換コードは、 $d=1$ 、 $k=7$ 、 $m=2$ 、 $n=3$ の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン d の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0024】請求項27に記載の復調装置は、入力された符号を、変換テーブルに従って、データに変換する変換手段を備え、変換テーブルの変換コードは、 $d=1$ 、 $k=7$ 、 $m=2$ 、 $n=3$ の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン d の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0025】請求項30に記載の復調方法は、入力された符号を、変換テーブルに従って、データに変換する変換スレッツを含み、変換テーブルの変換コードは、 $d=1$ 、 $k=7$ 、 $m=2$ 、 $n=3$ の基礎コードと、データ列

の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランドの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0026】請求項31に記載の提供媒体は、基本符号長がnビットの可変長符号(d, k; m, n; r)を、基本データ長がmビットのデータに変換する復調装置に、入力された符号を、変換テーブルに従って、データに変換する変換スラップを含む処理を実行させるプログラムを提供する提供媒体であって、変換テーブルの変換コードは、d=1、k=7、m=2、n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランドの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0027】請求項1に記載の変調装置、請求項25に記載の変調方法、および請求項26に記載の提供媒体、並びに、請求項27に記載の復調装置、請求項30に記載の復調方法、および請求項31に記載の提供媒体においては、データ列の要素内の「1」の個数と、符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則の変換コード、最小ランドの連続を所定の回数以下に制限する第1の置き換えコード、および、ラン長制限を守るための第2の置き換えコードに基づいて、変換処理が行われる。

【0028】
【発明の実施の形態】以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態(但し一例)を付加して本発明の特徴を記述すると、次のようになる。但し勿論この記載は、各手段を記載したものに限定することとを意味するものではない。

【0029】請求項1に記載の変調装置は、入力されたデータを、変換テーブル(例えば、表2)に従って、符号に変換する変換手段(例えば、図1の変調部12)を備え、変換テーブルの変換コードは、d=1、k=7、m=2、n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、*

<表2>

17PP.RML.32	
データ	符号
11	*0*
10	001

*どちらも1あるいは0で一致するような変換規則と、最小ランドの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0030】請求項12に記載の変調装置は、変換テーブルの変換コードとして存在しないユニークなパターンを含む同期信号を、符号語列の任意の位置に挿入する挿入手段(例えば、図9の同期信号挿入部212)をさらに備えることを特徴とする。

【0031】請求項23に記載の変調装置は、入力されたデータのDSVを制御して、変換手段に供給するDSV制御手段(例えば、図1のDSV制御ビット決定・挿入部11)をさらに備えることを特徴とする。

【0032】請求項24に記載の変調装置は、変換手段は、最小ランドの連続を制限する第1の置き換えコードを検出する第1の検出手段(例えば、図3の最小ラン連続制限コード検出部33)と、ラン長制限を守る第2の置き換えコードを検出する第2の検出手段(例えば、図3の最大ラン補償コード検出部34)とを備えることを特徴とする。

【0033】請求項27に記載の復調装置は、入力された符号を、変換テーブル(例えば、表2)に従って、データに変換する変換手段(例えば、図5の復調部111)を備え、変換テーブルの変換コードは、d=1、k=7、m=2、n=3の基礎コードと、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ランドの連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードとを有することを特徴とする。

【0034】請求項28に記載の復調装置は、所定の間隔で挿入された冗長ビットを除去する除去手段(例えば、図5のDSV制御ビット除去部112)をさらに備えることを特徴とする。

【0035】次に、本発明の実施の形態について説明するが、以下においては、説明の便宜上、変換される前のデータの「0」と「1」の並び(変換前のデータ列)を、(000011)のように、()で区切って表示し、変換された後の符号の「0」と「1」の並び(符号語列)を、“000100100”のように、“ ”で区切って表すことにする。以下に示す表2及び表3は、本発明のデータを符号に変換する変換テーブルの例を表している。

【0036】

(7)

11	
01	010
0011	010 100
0010	010 000
0001	000 100
000011	000 100 100
000010	000 100 000
000001	010 100 100
000000	010 100 000
"110111	001 000 000(next 010)
00001000	000 100 100 100
00000000	010 100 100 100

if xxl then *0* = 000
xx0 then *0* = 101

Sync & Termination
#01 000 000 001 (12 channel bits)
or
#01 001 000 000 001 000 000 001 (24 channel bits)
= 0 not terminate case
= 1 terminate case
Termination table
00 000
0000 010 100
"110111 001 000 000(next010):
When next channel bits are '010',
convert '11 01 11' to '001 000 000' after
using main table and termination table.

【0037】表2の変換テーブルは、変換コードとして、それがないと変換処理ができない基礎コード（データ列（11）から（000000）までのコード）、それがないと変換処理は可能であるが、それがあると、より効果的な変換処理が可能となる置き換えコード（データ列（110111）、（00001000）、（00000000）のコード）、および、符号を任意の位置で終端させるための終端コード（データ列（00）、（0000）のコード）を含んでいる。また、この変換テーブルには、同期信号も規定されている。

【0038】また、表2は、最小ラウンド＝1、最大ラウンド＝7で、基礎コードの要素に不確定符号（*を含む符号）を含んでいる。不確定符号は、直前および直後の符号語列の如何によらず、最小ラウンドと最大ラウンドを守るように、“0”か“1”に決定される。すなわち表2において、変換する2ビットのデータ列が（11）であったとき、その直前の符号語列によつて、“000”または“101”が選択され、そのいずれかに変換される。すなわち、直前の符号語列の1チャネルビットが“1”である場合、最小ラウンドを守るために、2ビットのデータ（11）は、符号語“000”に変換され、直前の符号語列の1チャネルビットが“0”である場合、最大ラ

ウンドが守られるように、符号語“101”に変換される。

【0039】表2の変換テーブルの基礎コードは可変長構造を有している。すなわち、拘束長1＝1における基礎コードは、必要数の4つ（2^m＝2²＝4）よりも少ない3つ（*0*, 001, 010の3つ）で構成されている。その結果、データ列を変換する際に、拘束長1＝1だけでは変換出来ないデータ列が存在することになる。結局、表2において、全てのデータ列を変換するには（変換テーブルとして成り立つためには）、拘束長1＝3までの基礎コードを参照する必要がある。

【0040】また、表2の変換テーブルは、最小ラウンドの連続を制限する置き換えコードを持っているため、データ列が（110111）である場合、さらに後ろに続く符号語列が参照され、それが“010”であるとき、このデータ列は、符号語“001000000”に置き換えられる。また、このデータ列は、後ろに続く符号語列が“010”以外である場合、2ビット単位（（11）、（01）、（11））で符号語に変換されるので、符号語“*0*010*0*”に変換される。これによつて、データを変換した符号語列は、最小ラウンドの連続が制限され、最大でも6回までの最小ラウンド繰返し

となる。

【0041】さらに、表2の変換コードは、データ列の要素内の“1”の個数を2で割った時の余りと、変換される符号語列の要素内の“1”の個数を2で割った時の余りが、どちらも1あるいは0で同一（対応するいずれの要素も、“1”の個数が奇数または偶数）となるような変換規則を持っている。例えば、変換コードのうちのデータ列の要素（000001）は、“010100100”の符号語列の要素に対応しているが、それぞれの要素の“1”の個数は、データ列では1個、対応する符号語列では3個であり、どちらも2で割ったときの余りが1（奇数）で一致している。同様にして、変換コードのうちのデータ列の要素（000000）は、“0101000000”の符号語列の要素に対応しているが、それぞれ“1”の個数は、データ列では0個、対応する符号語列では2個であり、どちらも2で割ったときの余りが0（偶数）で一致している。

【0042】そして、表2では、最大拘束長 r = 4 である。拘束長 i = 4 のコードは、最大ランク k = 7 を実現するための、置き換えコード（最大ランク値コード）で構成されている。すなわち、データ（00001000）は、符号語“000100100100”に変換され、データ（00000000）は、符号語“010100100100100”に変換されるようになされている。なお、この場合にも、最小ランク = 1 は守られている。

【0043】この拘束長 i = 4 の置き換えコードを設けない時、表2は最大拘束長 r = 3 となり、最大ランク k が8の符号を作ることができる。しかし、この拘束長 i = 4 のコードを設けることで、最大ランクを7にすることができ。一般的に、最大ランク k が大きいほど、クロックの再生には不利となり、システムの安定性が悪くなる。従って、最大ランクを8から7にすることで、この特性がそれだけ改善されることになる。

【0044】以上より表2のテーブルにおいて、基礎コードのみによってテーブルを構成する場合は、最大拘束長 r = 3 となり、最小ランク = 1 で最大ランク k = 8、かつデータ列の要素内の“1”の個数を2で割った時の余りと、変換される符号語列の要素内の“1”の個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号が発生することができ。

【0045】また基礎コードに加えて、最小ランク d の連続を制限する置き換えコードを持つて構成する場合は、最大拘束長 r = 3 となり、最小ランク = 1 で最大ランク k = 8、かつ最小ランク d の連続が有制限までに制限され、さらにデータ列の要素内の“1”の個数を2で割った時の余りと、変換される符号語列の要素内の“1”の個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号が発生することができ。

【0046】そして、表2のように、基礎コードに加えて、最小ランク d の連続を制限する置き換えコードを持

ち、さらに最大ランク k を7に補償する置き換えコードを持つて構成する場合は、最大拘束長 r = 4 となり、最小ランク = 1 で最大ランク k = 7、かつ最小ランク d の連続が有制限までに制限され、さらにデータ列の要素内の“1”の個数を2で割った時の余りと、変換される符号語列の要素内の“1”の個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号が発生することができ。

【0047】また、一般的に、最大拘束長 r が大きいほど、ビットシフト時の復調エラー（エッジビットの位置が1ビット分だけ、正規の位置よりも前方または後方にシフトすることによるエラー）の伝搬特性が悪くなる。

【0048】但し、表1と表2を比較すると、表1の最大拘束長 r は2であるのに対して、表2の最大拘束長 r は4である。従って、表1より表2の方がこの特性が悪いはずである。しかしながら、表7を参照して、シミュレーション結果として後述するように、表2の場合のこの特性は、表1のそれに較べてそれ程、劣るものではなかった（表7に示すように、平均バイトエラーレートが、表1の場合、1. 014バイトであるのに対して、表2の場合、1. 167バイトとなっており、それ程大きくない）。これは、変換コードの対の数が2個と少ないためと考えられる。

【0049】ところで上記表2の変換テーブルによって発生された符号語列（チャネルビット列）中の、任意の位置に同期信号を挿入する場合、この変換テーブルは可変長構造を有しているために、任意の位置で符号を終端させるために終端用テーブルが規定され、必要に応じて用いられるようになされている。

【0050】例えば、任意の位置で同期信号を挿入する際、まず直前直後の符号語列との接続において、最小ランク d 及び最大ランク k が守られるように接続ビットが設定され、接続ビットの間に同期信号用のユニークなパターンが設定される。同期信号パターンとして、最大ランク = 7 を破るパターンを与えたとき、最も短い長さで実現できる同期信号パターンは、次に示すように、12符号語（12チャネルビット）である。

“ # 0 1 0 0 0 0 0 0 0 0 0 1 ”

この同期信号パターンの先頭の“ # ”は接続用ビットで、“0”か“1”のどちらかに設定される。“#”の次の第2チャネルビット目は、最小ランクを守るために“0”に設定される。第3チャネルビット目から、同期信号パターンとして、k = 8 となる9Tのユニークなパターン（表2には規定されていないコードのパターン）が設定される。すなわち“1”と“1”の間に、“0”が8個連続して並ぶ。なおこの同期信号パターンの最後のチャネルビットは“1”とされているが、表2の変換テーブルを用いた時は、このようにしても、最小ランク d を守ることができる。

【0051】次に終端用テーブルと、この同期信号パタ

ーシの接続用ビット”#”について説明する。終端用テーブルは、表2に示すように、

00 000
0000 010 100

となる。終端用テーブルが必要になるのは、データ列と符号語列の対の数が4つ ($2^m = 2^2 = 4$) よりも少ない拘束長 i の基礎コードに対してである。

【0052】すなわち、表2では、拘束長 i = 1における基礎コードのデータ列と符号語列の対の数は3つであるから終端用テーブルが必要となる。また拘束長 i = 2における基礎コードのデータ列と符号語列の対の数も3つであるから終端用テーブルが必要となる。拘束長 i = 3における変換コードのデータ列と符号語列の対の数は5つあり、そのうち1つが置き換えコードで、4つが基礎コードであり、必要数 (4個) を持っているので、この場合、終端用テーブルは不要となる。拘束長 i = 4におけるコードは、いずれも置き換えコードであるため、符号の終端を考慮しなくてよい。従って、終端用テーブルは、i = 1の (00) のときと、i = 2のときの (0000) のときに必要になる。この終端用テーブルにより、データ (00) は、符号”000”に変換され、データ (0000) は、符号”010100”に変換され*

10 *。これにより、同期信号を挿入するに際し、その直前のデータを符号に変換することができなくなる (同期信号の直前までの符号を終端させることができなくなる) ことが防止される。
【0053】同期信号パターンの接続用ビット”#”は、終端用のテーブルを用いる場合と、用いない場合を区別するためのものである。すなわち、同期信号として与えられた、先頭の第1チャネルビット目の”#”は、終端コードを用いたときは「1」とされ、そうでないときは「0」とされる。こうすることによって、データの違い (終端コードを用いたか否か) を、間違いなく識別することができ。

【0054】以上においては、同期信号パターンを最短である12符号語 (12チャネルビット) として説明したが、同期信号パターンとしては、最大ランク=7を破る、k=8 (9T) が作成できれば良いので、12符号語以上であれば同期信号パターンがこの他にも作れることになる。例えば15符号語ならば

”#0100000000001010”
”#01000000000001000000001”
※出することが可能となる。そして、24符号語の場合、

のように、k=8 (9T) のパターンを2回繰り返し返すことができ、このようにすると、同期信号をより確実に検出

”#010010010000000001000000000001”

のように、”3T-9T-9T”を同期信号パターンとし、2回連続するk=8 (9T) のパターンの前後に、大きなランシ (T) が来る確率を減らし、さらに検出能力を高くすることができ。どの程度の検出能力の同期信号を採用するかは、システムの要求によって、選択する★

＜表3＞

17PP.RML.52

i=1 Main table:

データ	符号
00	101
01	100
10	001
11	000

i=2 substitution table A. (limits d to 1)

0000	100 010
0001	101 010
1000	000 010
1001	001 010

i=3 substitution table B. (limits k to 8)

111111	000 010 010
111110	001 010 010
011110	101 010 010
011111	100 010 010

i=4 substitution table C. (limits RMTR to 6)

(10)

17

	00010001	100 010 010 010
chan --0	10010001	100 000 010 010
chan --1	10010001	000 010 010 010
r=4 substitution table D. (limits k to 7)		
chan 010	11100000	000 001 010 010
chan 010	11100010	100 001 010 010
chan 010	11100001	001 010 010 010
chan 010	11100011	101 010 010 010

Sync

data: x1.....0x

ch. : xx0 100 000 000 10x (12channel bits)

data: x1.....0x

ch. : xx0 100 000 000 100 000 000 10x (24channel bits)

Termination :

add data bits '01' or '11' at begin,

and '00' or '01' at the ent

【0057】表3は、最小ランク＝1、最大ランク＝7で、拘束長 i＝1において基礎コードを4つ(2^m＝2²＝4)持つ構造をしている。すなわち、表3では、拘束長 i＝1の場合がメインテーブルとされ、拘束長 iが、2以上の場合は、d, k等を制限するための置き換えコードのテーブルとなっている。すなわち、i＝2のテーブルAでは、最小ランクを1に制限する置き換えコードが規定されており、i＝3のテーブルBでは、最大ランクを8までに制限する置き換えコードが規定されており、さらに、i＝4のテーブルC及びテーブルDでは、最小ランク＝1の連続を制限する置き換えコード(テーブルC)と、最大ランクを7までに制限する置き換えコード(テーブルD)とが規定されている。表3では、最大拘束長 r＝4である。

【0058】このように、表3は、変換テーブル内に、最小ランクの連続を制限する置き換えコードを持っているので、例えば、データ列(00010001)は、符号語列"100010010010"に置き換えられる。また、データ列(10010001)は、その直前の符号語列を参照して、それが"0"か"1"かによって置き換えコードが選択され、"0"ならば、"100000010010"に、"1"ならば、"000010010010"に、それぞれ変換される。これによって、データ変換後の符号語列は、最小ランクの連続が制限され、最大でも6回までの最小ランク繰り返しの符号語列となる。

【0059】また、表3は、データ列の要素内の「1」の個数を2で割ったときの余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも、1あるいは0で、同一となるような変換規則を持っている。例えば、データ列の要素(1000)は、"000010"の符号語列に対応しているが、それぞれ

の「1」の個数は、データ列では1個(奇数)、対応する符号語列では1個(奇数)であり、どちらも2で割ったときの余りが1(奇数)で一致している。同様に、データ列の要素(111111)は"000010010"の符号語列に対応しているが、それぞれの"1"の個数は、データ列では6個(偶数)、対応する符号語列では2個(偶数)であり、どちらも2で割ったときの余りが0(偶数)で一致している。

【0060】さらに、表3は、最大拘束長である r＝4のコードにおいて、最大ランク＝7を実現するための置き換えコードを持つ。このとき、その直前の符号語列が参照され、それが、"010"であるとき置き換えが行われる。例えば、データ(11100000)は、直前の符号語が"010"であれば、符号語"00001010010"に変換され、データ(11100010)は、直前の符号語が"010"であれば、符号語"100001010010"に変換される。

【0061】以上より表3のテーブルにおいて、RL符号を実現するためには、基礎コードのみによってテーブルを構成することはできない。

【0062】最小ランクおよび最大ランクを補償するために、基礎コードおよび、テーブルA(拘束長 i＝2)、テーブルB(i＝3)を用いることで、RL符号を作成することができる。この場合は、最大拘束長 r＝3となり、最小ランク＝1で最大ランク＝8、かつデータ列の要素内の"1"の個数を2で割った時の余りと、変換される符号語列の要素内の"1"の個数を2で割った時の余りが、どちらも1あるいは0で同一となるような、符号を発生することができる。

【0063】また基礎コード、テーブルA及びテーブルBに加えて、最小ランクの連続を制限する置き換えコード(テーブルC)を持って構成する場合は、最大拘束長

r = 4 となり、最小ラウンド = 1 で最大ラウンド = 8、かつ最小ラウンドの連続が有限回までに制限され、さらにデータ列の要素内の“ 1 ”の個数を 2 で割った時の余りと、変換される符号語列の要素内の“ 1 ”の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で同一となるような、符号を発生することができる。なおこの場合では、表 3 内のテーブル C のような、直前の符号語列を参照することは、必ずしも必要ではない。

【0064】そして、表 3 のように、基礎コード、テーブル A、B に加えて、最小ラウンドの連続を制限する置き換えコード（テーブル C）を持ち、さらに最大ラウンドに補償する置き換えコード（テーブル D）を持つて構成する場合は、最大拘束長 r = 4 となり、最小ラウンド = 1 で最大ラウンド k = 7、かつ最小ラウンドの連続が有限回までに制限され、さらにデータ列の要素内の“ 1 ”の個数を 2 で割った時の余りと、変換される符号語列の要素内の“ 1 ”の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で同一となるような、符号を発生することができる。

【0065】上記表 3 の変換テーブルを利用して発生させた符号語列（チャネルビット列）中の任意の位置に同期信号を挿入する場合、この変換テーブルは拘束長 i = 1 において終端が可能であるから、表 2 で述べたような終端用テーブルは不要である。

【0066】そして、同期信号パターンをなるべく効率良く挿入するために、以下のように同期信号パターンを決定する。すなわち同期信号としてのユニークなパターン（符号語列）の前後の 3 符号語を、表 3 に従って変換された符号語で構成されるようにする。直前直後の 3 ビットは、以下に示すように、データビットと接続ビットが混合した形式となっている。

【0067】すなわち、直前の 3 ビット（符号語）は、変換前のデータ語において、m ビット（2 ビット）単位で見て、先頭の 1 ビット目を情報データ語とし、次の 2 ビット目を同期信号を規定するために「1」とし、これ*

” x x 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 x ”

のように、k = 8（9 T）のパターンを 2 回繰り返すパターンを同期信号パターンとすることができる。このような同期信号パターンにすれば、同期信号検出能力を強めることができる。どの程度の検出能力のパターンを同期信号パターンとするかは、システムの要求によって、選択することが出来る。

【0071】表 2 及び表 3 のような変換テーブルを用いた時、従来の場合と同様に、データ列を変調した後、変調後のチャネルビット列に、所定の間隔で、DSV 制御ビットを付加することで、DSV 制御することができる。しかしながら、表 2 および表 3 においては、データ列と、変換される符号語列の関係を生かして、さらに効率良く DSV 制御を行うことができる。

【0072】即ち、変換テーブルが、データ列の要素内

* を変換テーブル表 3 において符号語（チャネルビット）に変換したものとする。このとき変換前の m ビット（2 ビット）のデータ語（x 1）は、n ビット（3 ビット）の符号語” x x 0 ”に変換される。

【0068】また直後の 3 ビット（符号語）は、変換前のデータ語において、m ビット（2 ビット）単位で見ても、先頭の 1 ビット目を同期信号を規定するために「0」とし、次の 2 ビット目を情報データ語とする。そして、この 2 ビットのデータ語を変換テーブル 3 により符号語（チャネルビット）に変換する。このとき、m ビット（2 ビット）のデータ語（0 x）は、n ビット（3 ビット）のデータ語” 1 0 x ”に変換される。

【0069】同期信号のユニークパターンを、最大ラウンド k = 7 を破るパターンとしたとき、最も短い長さで実現できる同期信号パターンは、次に示すような 1 2 符号語（1 2 チャネルビット）である。

” x x 0 1 0 0 0 0 0 0 0 0 1 0 x ”

” x ”の値は変換テーブルに依存する。上の 1 5 符号語の中には、2（=先頭の 1 ビット+最後の 1 ビット）データ語すなわち 3 符号語相当を含んでいるので、実際には 1 2 符号語が同期信号パターンのための冗長な部分である。第 3 チャネルビット目は、最小ラウンドを守るために” 0 ”が設定される。第 4 チャネルビット目から、同期信号パターンとして、k = 8 となる 9 T が設定される。すなわち” 1 ”と” 1 ”の間に、” 0 ”が 8 個連続して並ぶ。

【0070】以上においては、同期信号パターンを最短である 1 2 符号語（1 2 チャネルビット）として説明したが、同期信号パターンとしては、最大ラウンド k = 7 を破る、k = 8（9 T）が作成できれば良いので、1 2 符号語以上であれば、この他にも、同期信号パターンが作ることが可能である。例えば 1 5 符号語ならば、

” x x 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 x ”

のような同期信号を作ることができる。また、2 1 符号語ならば、

” x x 0 0 0 0 0 0 0 1 0 x ”

の” 1 ”の個数と、変換される符号語列の要素内の” 1 ”の個数を、2 で割った時の余りが、どちらも 1 あるいは 0 で同一となるような変換規則を持っている時、上記のようにチャネルビット列内に、「反転」を表す” 1 ”、あるいは「非反転」を表す” 0 ”の DSV 制御ビットを挿入することは、データビット列内に、「反転」するならば（1）の、「非反転」ならば（0）の、それぞれ DSV 制御ビットを挿入することと等価となる。

【0073】たとえば、表 2 において、データ変換する 3 ビットが（0 0 1）と続いた時に、その後ろにおいて DSV 制御ビットを挿入するものと、データは、（0 0 1ーx）（x は 1 ビットで、「0」又は「1」）となる。ここで x に「0」を与えれば、表 2 の変換テーブルで、

データ列 符号語列
0010 010 000
の変換が行われ、また、「1」を与えれば、
データ列 符号語列
0011 010 100

の変換が行われる。符号語列をNRZI化して、レベル符号化したとき、これらは

データ列 符号語列 レベル符号列
0010 010 000 011111
0011 010 100 011000

となり、レベル符号列の最後の3ビットが相互に反転している。このことは、DSV制御ビットxの、(1)と(0)を選択することによって、データ列内においても、DSV制御が行えることを意味する。

【0074】DSV制御による冗長度を考えると、データ列内の1ビットでDSV制御を行うことは、チャネルビット列で表現すれば、表2及び表3の変換率($m/n = 2/3$)より、1.5チャネルビットでDSV制御を行っていることに相当する。例えば、表1のようなRL(1-7)テーザルにおいてDSV制御を行うためには、チャネルビット列においてDSV制御を行う必要があるが、この場合、最小ランシフトを守るためには、少なくとも2チャネルビットが必要であり、表2と表3においてデータ語でDSV制御するのに比較して、冗長度がより大きくなってしまふ。換言すれば、本方式のように、データ列内でDSV制御を行うことで、効率よくDSV制御を行うことができる。

【0075】次に、図1を参照して、本発明に係る変調装置の実施の形態を図面を参照しながら説明する。この実施の形態では、データ列が、表2に従って、可変長符号(d, k ; m, n ; r) = (1, 7 ; 2, 3 ; 4)に変換される。

【0076】図1に示すように、変調装置1は、DSV制御ビットである「1」あるいは「0」を決定し、入力されたデータ列に、任意の間隔で挿入するDSV制御ビット決定・挿入部11、DSV制御ビットが挿入されたデータ列を変調する変調部12、並びに、変調部12の出力を記録波形列に変換するNRZI化部13を備える。また、変調装置1は、タイミング信号を生成し、各部に供給してタイミングを管理するタイミング管理部14を備える。

【0077】図2は、DSV制御ビット決定・挿入部11の処理を説明する図である。DSV制御ビットの決定及び挿入は、データ列内の任意の間隔おきに行われる。図2に示すように、入力されたデータ語のうち、まずDATA1とDATA2の間にDSV制御ビットを挿入するために、DSVビット・決定挿入部11は、DATA1までの積算DSVを計算する。DSV値は、DATA1を、チャネルビット列に変換し、さらにレベル符号化(NRZI化)した各レベルを、レベルH(1)を「+1」、レベルL(0)を「-1」として、それらの値を積算することによって得られる。同様に、

次の区間DATA2における区間DSVが計算される。次に、DATA1とDATA2の間に挿入されるDSV制御ビットx1として、DATA1、DSV制御ビットx1、およびDATA2によるDSVの絶対値が「ゼロ」に近づくような値を決定する。

【0078】DSV制御ビットx1を(1)に設定すると、DATA1の後の区間DATA2のレベル符号は反転され、また、(0)に設定すると、DATA1の後の区間DATA2のレベル符号は非反転となる。なぜならば、上記表2及び表3の各テーザル内の要素は、データ列の要素内の「1」の個数と、変換される符号語列の要素内の「1」の個数を、2で割った時の余りが、どちらも1あるいは0で一致するようになっているので、データ列内において、(1)を挿入することは、すなわち、変換される符号語列に“1”を挿入することになる(すなわち「反転」されることになる)からである。

【0079】このようにして、図2のDSV制御ビットx1が決定したら、次に所定のデータ間隔において、DATA2とDATA3の間に、DSV制御ビットx2を挿入し、同様にDSV制御を行う。なおそのときの積算DSV値は、DATA1, x1, そしてDATA2までの全てのDSV値とする。

【0080】このように、あらかじめデータ列内に、DSV制御ビットが挿入された後、変調部12で変換が行なわれ、チャネルビット列が発生される。

【0081】図3は、変調部12の構成例を示すブロック図である。図3において、シフトレジスタ31は、データを2ビットずつシフトさせながら、拘束長判定部32、最小ラン連続制限コード検出部33、ラン長制限補償コード検出部34、および全ての交換部35-1乃至35-4に出力するようになされている。このときシフトレジスタ31は、各部がその処理を行うのに必要なビット数を各部に供給する。

【0082】拘束長判定部32は、データの拘束長iを判定し、マルチプレクサ36に出力するようになされている。最小ラン連続制限コード検出部33は、最小ランの連続を制限する専用のコード(表2の場合、(110111))を検出したとき、その拘束長を表す検出信号(i=3)を拘束長判定部32に出力する。またラン長制限補償コード検出部34は、表2では最大ランを補償する専用のコード(表2の場合、(000010000))、または、(00000000))を検出したとき、その拘束長を表す検出信号(i=4)を拘束長判定部32に出力する。

【0083】最小ラン連続制限コード検出部33により専用のコードが検出されたとき、あるいはラン長制限補償コード検出部34により、専用のコードが検出されたとき、拘束長判定部32は、対応する拘束長をマルチプレクサ36に出力する。このとき、拘束長判定部32では、別の拘束長を判定している場合があるが、最小ラン連続制限コード検出部33またはラン長制限補償コード検出部34から専用コードによる検出出力があれば、拘

束長判定部 3 2 は、そちらを優先させて拘束長を決定する。言い換えれば、より大きい拘束長が選択される。

【0 0 8 4】変換部 3 5ー1 乃至 3 5ー4 は、内蔵されている変換テーブルを参照し、供給されたデータに対応する基礎コードが登録されているか否かを判断し、登録されている場合は、そのデータに対応する符号語に変換した後、変換後の符号語をマルチプレクサ 3 6 に出力するようになされている。また、対応するデータが変換テーブルに基礎コードとして登録されていない場合、変換部 3 5ー1 乃至 3 5ー4 は、入力されたデータを破壊するようになされている。

【0 0 8 5】なお、この変調装置 1 2 は、表 2 に対応するもので、変換部 3 5ー1 としては、拘束長 i = 4 までのものが用意されている。すなわち、変換部としては、最大拘束長 r までの分が用意されることになる。

【0 0 8 6】マルチプレクサ 3 6 は、拘束長判定部 3 2 より供給される拘束長 i に対応する変換部 3 5ーi が変換した符号を選択し、その符号を、シリアルデータとして、バスツェ 3 7 を介して出力するようになされている。

【0 0 8 7】また各部の動作のタイミングは、タイミング管理部 1 4 から供給されるタイミング信号に同期して管理されている。

【0 0 8 8】次に、この実施の形態の動作について説明する。

【0 0 8 9】最初に、シフトレジスタ 3 1 より、各変換部 3 5ー1 乃至 3 5ー4、拘束長判定部 3 2、最小連続制限コード検出部 3 3、およびラン長制限補償コード検出部 3 4 に、データが、2 ビット単位で、それぞれが判定等に必要なビット数だけ供給される。

【0 0 9 0】拘束長判定部 3 2 は、例えば表 2 に示す変換テーブルを内蔵しており、この変換テーブルを参照して、データの拘束長 i を判定し、判定結果 (拘束長 i) をマルチプレクサ 3 6 に出力する。

【0 0 9 1】最小ラン連続制限コード検出部 3 3 は、表 2 に示す変換テーブルのうちの、最小ランの連続を制限する置き換えコード (表 2 の場合、データ (1 1 0 1 1 1) と、後ろに続く符号語" 0 1 0 " を変換する部分) を内蔵しており、この変換テーブルを参照して、最小ランの連続を制限するコードを検出したとき、拘束長 i = 3 の検出信号を拘束長判定部 3 2 に出力する。

【0 0 9 2】またラン長制限補償コード検出部 3 4 は、表 2 に示す変換テーブルの中の、最大ランを守る置き換えコード (表 2 の場合、データ (0 0 0 0 1 0 0 0) および (0 0 0 0 0 0 0 0)) を内蔵しており、この変換テーブルを参照して、最大ランを守る置き換えコードを検出したとき、拘束長 i = 4 の検出信号を拘束長判定部 3 2 に出力する。

【0 0 9 3】拘束長判定部 3 2 は、最小ラン連続制限コード検出部 3 3 から拘束長 i = 3 の検出信号が入力され

た場合には、そのとき別の拘束長を判定していたとしても、それを選択せず、最小ラン連続制限コード検出部 3 3 の検出に対応する拘束長 i (表 2 の例の場合、i = 3) をマルチプレクサ 3 6 に出力する。同様に拘束長判定部 3 2 は、ラン長制限補償コード検出部 3 4 から拘束長 i = 4 の検出信号が入力された場合には、そのとき別の拘束長を判定していたとしても、それを選択せず、ラン長制限補償コード検出部 3 4 の検出に対応する拘束長 i (表 2 の例の場合、i = 4) をマルチプレクサ 3 6 に出力する。

【0 0 9 4】このことは、結局、各検出部 3 3、3 4 における拘束長の判定結果と、判定部 3 2 における拘束長の判定結果が、異なった場合には、大きい方の拘束長を最終的な拘束長として選択すればよいことを意味する。

【0 0 9 5】図 4 は、拘束長判定部 3 2、最小ラン連続制限コード検出部 3 3、及びラン長制限補償コード検出部 3 4 の動作の具体例を説明している。

【0 0 9 6】ラン長制限補償コード検出部 3 4 は、表 2 に示すテーブルの、(0 0 0 0 1 0 0 0) および (0 0 0 0 0 0 0 0) の変換部分を持ち、入力された 8 ビットのデータが、これと一致する場合、拘束長 i = 4 の検出信号を拘束長判定部 3 2 に出力する。

【0 0 9 7】最小ラン連続制限コード検出部 3 3 は、表 2 に示すテーブルの、データ (1 1 0 1 1 1) と符号" 0 1 0 " の変換部分を持ち、入力された 6 ビットのデータが、(1 1 0 1 1 1) であり、その後の 3 符号語が、" 0 1 0 " である場合、拘束長 i = 3 の検出信号を拘束長判定部 3 2 に出力する。なお、3 符号語" 0 1 0 " の部分を、データ変換前のデータ列で表せば、(0 1), (0 0 1) 又は (0 0 0 0 0) となる。従って最小ラン連続制限コード検出部 3 3 は、言い換えれば、(1 1 0 1 1 1) + (0 1 / 0 0 1 / 0 0 0 0 0) の変換部分を持ち、入力された 6 ビットのデータに加えて、その後の 5 ビットのデータまでをさらに参照し、それらがこれらのいずれかと一致する場合 ((1 1 0 1 1 1 0 1), (1 1 0 1 1 1 0 0 1) または (1 1 0 1 1 1 0 0 0 0) のいずれかである場合)、拘束長 i = 3 の検出信号を拘束長判定部 3 2 に出力する。

【0 0 9 8】また拘束長判定部 3 2 は、表 2 に示すテーブルの変換コードを内蔵しており、入力された 6 ビットのデータが、(0 0 0 0 1 1), (0 0 0 0 1 0), (0 0 0 0 0 1), あるいは (0 0 0 0 0 0) のいずれかに一致する場合、拘束長 i = 3 と判定する。また、入力された 4 ビットのデータが (0 0 1 1), (0 0 1 0), (0 0 0 1) のいずれかに一致する場合、拘束長判定部 3 2 は、拘束長 i = 2 と判定する。さらに入力された 2 ビットのデータが (1 1), (1 0), (0 1) のいずれかに一致する場合、拘束長判定部 3 2 は、拘束長 i = 1 と判定する。

【0 0 9 9】ところで、入力されたデータが例えば (0

送路より伝送されてきた信号、または、記録媒体より再生された信号を、復調テークル（逆変換テークル）に基づいて復調する復調部 1 1 1、並びに、復調されたデータ列より、任意の間隔で挿入されているデータ列内のDSV制御ビットを除去し、元のデータ列を復元するDSV制御ビット除去部 1 1 2を備える。パツファ 1 1 3は、DSV制御ビット除去部 1 1 2から入力されたシリアルデータを一旦記憶し、所定の転送レークで読み出し、出力する。タイミツダ管理部 1 1 4は、タイミツダ信号を生成し、各部に供給してタイミツダを管理する。

【0 1 1 2】復調部 1 1 1は、図 6に示すように、伝送路より伝送されてきた信号、または、記録媒体より再生された信号を 2 値化するコンパレーク部 1 2 1を備える。コンパレーク部 1 2 1はまた、入力された信号がNRZI変調されている時（レベル符号である時）、これを逆NRZI符号化（エツジ符号化）する。拘束長判定部 1 2 2は、コンパレーク部 1 2 1によりデジタル化された信号の入力を受け、拘束長 i を判定する。また最小ラン連続制限コード検出部 1 2 3は、コンパレーク部 1 2 1より入力されたデジタル化された信号から、最小ランの連続を制限するために与えられた専用のコード（表 2 の“0 0 1 0 0 0 0 0”）を検出し、それに対応する拘束長 i = 3 の検出信号を拘束長判定部 1 2 2に送る。さらにラン長制限補償コード検出部 1 2 4は、コンパレーク部 1 2 1より入力された信号から、最大ランを補償するために与えられた専用のコード（表 2 の“0 0 0 1 0 0 1 0 0 1 0 0”、“0 1 0 1 0 0 1 0 0 1 0 0”）を検出し、それに対応する拘束長 i = 4 の検出信号を拘束長判定部 1 2 2に送る。

【0 1 1 3】逆変換部 1 2 5 - 1乃至 1 2 5 - 4は、n x i ビットの可変長符号を、m x i ビットのデータに逆変換するテークル（表 2 の場合、i = 1乃至 4 のテークルで、変換部 3 5 - 1乃至 3 5 - 4 の変換テークルと実質的に同一の変換テークル）を有している。マルチプレクサ 1 2 6は、逆変換部 1 2 5 - 1乃至 1 2 5 - 4 の出力のいずれかを、拘束長判定部 1 2 2 の判定結果に対応して選択し、シリアルデータとして出力する。

【0 1 1 4】次に図 6 の復調部 1 1 1 の動作について説明する。伝送路より伝送されてきた信号、あるいは記録媒体より再生された信号は、コンパレーク部 1 2 1に入力され、コンパレークされる。コンパレーク部 1 2 1より出力された信号は、逆NRZI符号（“1”がエツジを示す符号）のデジタル信号となつて、拘束長判定部 1 2 2に入力され、表 2 に示す変換テークル（逆変換テークル）に従つて、拘束長の判定処理が行われる。拘束長判定部 1 2 2 の判定結果（拘束長）はマルチプレクサ 1 2 6に出力される。

【0 1 1 5】コンパレーク部 1 2 1から出力されたデジタル信号は最小ラン連続制限コード検出部 1 2 3にも入力される。最小ラン連続制限コード検出部 1 2 3は、表

2 に示す、変換テークルのうちの、最小ランの連続を制限する置き換えコード（表 2 の場合、符号語“0 0 1 0 0 0 0 0 0”を変換する部分）を内蔵しており、この逆変換テークルを参照して、最小ランの連続を制限するコード“0 0 1 0 0 0 0 0 0 not 1 0 0”を検出したとき、拘束長 i = 3 の検出信号を拘束長判定部 1 2 2に出力する。

【0 1 1 6】さらに、コンパレーク部 1 2 1から出力されたデジタル信号はラン長制限補償コード検出部 1 2 4にも入力される。ラン長制限補償コード検出部 1 2 4は、表 2 に示す、変換テークルの中の、最大ランを守る置き換えコード（表 2 の場合、符号語列“0 0 0 1 0 0 1 0 0 1 0 0”、“および”0 1 0 1 0 0 1 0 0 1 0 0”）を内蔵しており、この逆変換テークルを参照して、最大ランを守る置き換えコードを検出したとき、拘束長 i = 4 の検出信号を拘束長判定部 1 2 2に出力する。

【0 1 1 7】入力された変調符号の判定処理についてまとめると、図 7 に示すようになる。すなわち、ラン長制限補償コード検出部 1 2 4は、表 2 に示すテークルの、“0 0 0 1 0 0 1 0 0 1 0 0”、あるいは“0 1 0 1 0 0 1 0 0 1 0 0”の逆変換部分を持ち、入力された 1 2 ビットの符号語列が、これと一致する場合、拘束長 i = 4 の検出信号を拘束長判定部 1 2 2に出力する。

【0 1 1 8】最小ラン連続制限コード検出部 1 2 3は、表 2 に示すテークルの、“0 0 1 0 0 0 0 0 0”の逆変換部分を持ち、入力された 1 2 ビットの符号語列が、“0 0 1 0 0 0 0 0 0 not 1 0 0”と一致する場合、拘束長 i = 3 の検出信号を拘束長判定部 1 2 2に出力する。なお、拘束長の判定には特に必要ないが、入力された符号語列を 1 2 ビット分見ると、このときの符号語は、“0 0 1 0 0 0 0 0 0 0 1 0”となっている。

【0 1 1 9】また拘束長判定部 1 2 2は、表 2 に示す逆変換テークルを内蔵しており、入力された 9 ビット又は 1 2 ビットの符号語列が、“0 0 0 1 0 0 1 0 0”、“0 0 0 1 0 0 0 0 0 not 1 0 0”、“0 0 0 1 0 0 0 0 0 not 1 0 0”、“0 1 0 1 0 0 1 0 0”、あるいは“0 1 0 1 0 0 0 0 0 not 1 0 0”のいずれかに一致するとき、拘束長判定部 1 2 2は、拘束長 i = 2 と判定する。さらにこれに当てはまらない場合、入力された 3 ビットの符号語列が、“0 0 0”、“1 0 1”、“0 0 1”、あるいは“0 1 0”のいずれかに一致するとき、拘束長判定部 1 2 2は、拘束長 i = 1 と判定する。

【0 1 2 0】なお、拘束長判定部 1 2 2、最小ラン連続

30

20

10

40

50

制限コード検出部 1 2 3、及びラン長制限補償コード検出部 1 2 4 の拘束長判定の処理は、拘束長の小さい方から、i = 1, i = 2, i = 3, i = 4 の順番で行うようにしてもよい。

【0 1 2 1】拘束長を、その小さい方から、i = 1, i = 2, i = 3, i = 4 の順番で判定していった場合、入力された符号語列が例えば、“0 0 0 1 0 0 1 0 0 1 0 0”であったとき、拘束長判定部 1 2 2 において、拘束長の小さいほうから順に、一致または不一致を判定していくと、拘束長 i = 1 あるいは、拘束長 i = 2、拘束長 i = 3、そして拘束長 i = 4 と、全ての拘束長にあてはまることになる。このような場合は、決定規則として、それぞれ判定された拘束長から最大のものを選択し、決定すればよい。

【0 1 2 2】逆変換部 1 2 5 - 1 乃至 1 2 5 - 4のうち、例えば逆変換部 1 2 5 - 1 には、“アドレス” 1 0 *

<表 4>

逆変換テーブル (1, 7 ; 2, 3 ; 4)

符号語列	復調データ列
i=1 101	11
000	11
001	10
010	01
i=2 010 100	0011
010 000(not 100)	0010
000 100	0001
i=3 000 100 100	000011
000 100 000(not 100)	000010
010 100 100	000001
010 100 000(not 100)	000000
i=3 : Prohibit Repeated Minimum Transition Runlength	
001 000 000(not 100)	110111
i=4 : limits k to 7	
000 100 100 100	00001000
010 100 100 100	0 0 0 0 0 0 0 0

【0 1 2 6】次に、図 8 のフローチャートを参照して、DSV制御ビット除去部 1 1 2 の動作について説明する。DSV制御ビット除去部 1 1 2 は、内部にカウンタを有しており、ステップ S 1 において、復調部 1 1 1 よりデータ列のビットが入力されると、その数をカウンタする。ステップ S 2 において、カウンタ値がDSV制御ビットを挿入する所定のデータ間隔に達したか否かが判定され、任意のデータ間隔ではないと判定された場合、ステップ S 3 において、復調部 1 1 1 より入力されたデータがそのままバッファ 1 1 3 に出力される。これに対して、ステップ S 2 において、所定のデータ間隔であると判定された場合、そのビットはDSV制御ビットであるから、ステップ S 3 の処理はスキップされる。すなわち、この場合には、そのビットはバッファ 1 1 3 に出力されず、廃棄される。

* 1 “および” 0 0 0 “にデータ (1 1) が、アドレス” 0 0 1 “にデータ (1 0) が、そしてアドレス” 0 1 0 “にデータ (0 1) が、それぞれ書き込まれている。以下、逆変換部 1 2 5 - 2 乃至 1 2 5 - 4 の各逆変換テーブルも、同様に、それぞれ対応するデータが書き込まれており、供給された 3 × i ビットの符号語列を、2 × i ビットのデータ列に変換し、そのデータ語をマルチプレクサ 1 2 6 に出力する。

【0 1 2 3】マルチプレクサ 1 2 6 は、逆変換部 1 2 5 - 1 乃至 1 2 5 - 4 より供給されたデータのいずれかを、拘束長判定部 1 2 2 の拘束長判定結果に対応して選択し、シリアルデータとして出力する。

【0 1 2 4】表 2 の逆変換テーブルを示すと、次の表 4 のようになる。

【0 1 2 5】

【0 1 2 7】次に、ステップ S 4 に進み、次のデータを入力する処理が実行される。そして、ステップ S 5 において、全てのデータに対する処理が終了したか否かが判定され、まだ処理していないデータが存在する場合には、ステップ S 1 に戻り、それ以降の処理が繰り返し実行される。ステップ S 5 において、全てのデータを処理したと判定された場合、処理は終了される。

【0 1 2 8】以上のようにして、DSV制御ビット除去部 1 1 2 より出力されるデータからは、DSV制御ビットが除去されることになる。このデータは、バッファ 1 1 3 を介して出力される。

【0 1 2 9】以上においては、復調装置 1 0 0 に表 2 の変換テーブル (表 4 の逆変換テーブル) を用いた場合について説明したが、表 3 の変換テーブル (表 5 に示す表 3 に対応する逆変換テーブル) を用いた場合にも、同様

の処理を実行することができる。この場合、図 6 の最小ラン連続制限コード検出部 1 2 3 には、表 3 における拘束長 i = 4 のテーブル C を与えれば良い。また、ラン長制限補償コード検出部 1 2 4 には、表 3 における拘束長 i = 2 のテーブル A、拘束長 i = 3 のテーブル B、及び拘束長 i = 4 のテーブル D を与えれば良い。

【0 1 3 0】
＜表 5＞

逆変換テーブル (1, 7 ; 2, 3 ; 4)

符号語列 復調データ列

r=1 Main table:

101	00
100	01
001	10
000	11

10

r=2 substitution table A. (limits d to 1)

100 010	0000
101 010	0001
000 010	1000
001 010	1001

20

r=3 substitution table B. (limits k to 8)

000 010 010	11111
001 010 010	11110
101 010 010	01110
100 010 010	01111

r=4 substitution table C. (limits RMT R to 6)

100 010 010 010	00010001
100 000 010 010	10010001
000 010 010 010	10010001

r=4 substitution table D. (limits k to 7)

000 001 010 010	11100000
100 001 010 010	11100010
001 010 010 010	11100001
101 010 010 010	11100000

30

【0 1 3 1】ところで、データに同期信号 (Sync) *
” # 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1

と決定する。” # ” は、同期信号の挿入により区切られた、直前のデータ列 (DSV制御ビットは含んで良い) に依存しており、区切られたデータ列を変換テーブルに従って変調した際に、終端テーブルを用いて終端させた場合には

40

” # ” = ” 1 ”
とされ、また終端テーブルを用いずに、表 2 のテーブルにより終端した場合には
” # ” = ” 0 ”

とされる。変調部 1 2 は、終端テーブルを用いた場合には、” # ” = ” 1 ” を、用いない場合には、” # ” = ” 0 ” を、同期信号決定部 2 1 1 に出力する。同期信号決定部 2 1 1 は、変調部 1 2 から、この” # ” の値の入力を受けると、これを同期信号の先頭ビットに挿入する。

50

*を挿入する必要がある場合がある。次に、この場合の変調装置 1 と復調装置 1 0 0 について、図 9 と図 1 0 を参照して説明する。これらの実施の形態でも、データ列が、表 2 に従って、可変長符号 (d, k ; m, n ; r) = (1, 7 ; 2, 3 ; 4) に変調され、また復調されるものとする。

【0 1 3 2】所定の間隔で同期信号を挿入する変調装置 1 においては、図 9 に示すように、DSV制御ビット決定・挿入部 1 1 の出力が、同期信号決定部 2 1 1 に供給される。同期信号決定部 2 1 1 にはまた、変調部 1 2 の出力も供給されている。同期信号決定部 2 1 1 は、入力された信号から同期信号を決定すると、その出力を同期信号挿入部 2 1 2 に出力している。同期信号挿入部 2 1 2 は、変調部 1 2 より入力される変調信号に、同期信号決定部 2 1 1 より入力される同期信号を挿入し、NRZI化部 1 3 に出力している。その他の構成は、図 1 における場合と同様である。

【0 1 3 3】同期信号決定部 2 1 1 は、同期信号パターンの 2 4 符号語とするとき、表 2 に従って、同期信号を、

0 0 0 0 0 0 0 0 0 1”
そして、その同期信号を同期信号挿入部 2 1 2 に出力する。

【0 1 3 4】同期信号挿入部 2 1 2 は、同期信号決定部 2 1 1 から入力される同期信号を、変調部 1 2 の出力に挿入し、NRZI化部 1 3 に出力する。その他の動作は、図 1 における場合と同様である。

【0 1 3 5】同期信号が挿入された後の最初のデータは、その先頭から (同期信号の直前のデータを考慮することなく) 変換処理される。変調部 1 2、および同期信号決定部 2 1 1 は、同期信号が挿入される所定の間隔をカウントするためのカウンタを備え、そのカウント値に対応して、同期信号の位置を決定する。

【0 1 3 6】なお、図 9 の例では表 2 の変換テーブルを用いるようにしたが、表 3 の変換テーブルを用いること

33

も可能である。この場合、例えば図 9 における同期信号決定部 2 1 1 は、表 3 の同期信号パターンとして、1 2 符号語の同期信号を採用するとき、

“ x x 0 1 0 0 0 0 0 0 0 0 1 0 x ”

を同期信号とする。“ x ”は、同期信号挿入により区切られた、直前及び直後のデータ列 (DSV制御ビットは含んで良い) に依存しており、先頭の 3 符号語と最後の 3 符号語は、表 3 により決定される。すなわち同期信号の挿入により区切られた最後のデータ列を (p)、また、その直後の先頭のデータ列を (q) とすると、(p 1) として、表 3 を用いて変換し、その後には “ 1 0 0 0 0 0 0 0 0 0 ” を挟み、最後に (0 q) として、表 3 を用いて変換すること、同期信号として必要な最大ランクを破る、k = 8 (9 T) を必ず発生することができる。

【0 1 3 7】図 1 0 は、図 9 の変調装置 1 により変調された符号を復調する復調装置 1 0 0 の構成例を表している。この例においては、所定の伝送経路を介して入力された符号が、復調部 1 1 1 と同期信号識別部 2 2 1 に入力されている。同期信号識別部 2 2 1 は、入力された符号と復調部 1 1 1 から入力された信号を用いて、同期信号を識別し、識別信号を同期信号除去部 2 2 2 に出カしている。同期信号除去部 2 2 2 は、復調部 1 1 1 から入力された復調信号から、同期信号識別部 2 2 1 の出力に対応して同期信号を除去し、同期信号を除去した信号を DSV制御ビット除去部 1 1 2 に出カしている。その他の構成は、図 5 における場合と同様である。

【0 1 3 8】同期信号識別部 2 2 1 は、内蔵するカウンタで符号語をカウンタし、そのカウンタ値から所定の間隔で挿入されている同期信号の位置を決定する。同期信号パターンの位置が判明したとき、同期信号識別部 2 2 1 は、次に変調時に定めた “ # ” の部分を読み取る。即ち同期信号ビット部分の先頭ビットを読み取り、それを復調部 1 1 1 に出カする。復調部 1 1 1 は、先頭ビットが “ 1 ” であれば、その直前の符号の復調には、表 2 の終端テーブルを用いる。また先頭ビットが “ 0 ” であれば、復調部 1 1 1 は、その直前の符号の復調には、表 2 の変換コードのテーブルを用いる。これ以外の同期信号ビットは、情報を持たないビットであるから不要となる。

【0 1 3 9】同期信号識別部 2 2 1 は、同期信号を構成するビットを識別する識別信号を同期信号除去部 2 2 2 に出カする。同期信号除去部 2 2 2 は、復調部 1 1 1 より入力されたデータから、この識別信号によって指定された同期信号ビットだけを除去し、DSV制御ビット除去部 1 1 2 に出カする。

【0 1 4 0】なお、図 1 0 では表 2 の変換テーブルを用いるようにしたが、表 3 の変換テーブルを用いることもできる。この場合、例えば図 1 0 における同期信号識別部 2 2 1 は、所定の間隔で挿入されている同期信号の位

34

置をカウンタの値から決定する。同期信号パターンの位置が判明したとき、同期信号識別部 2 2 2 は、同期信号パターンの先頭の 3 符号語、及び、最後の 3 符号語を指定する信号を復調部 1 1 1 に出カする。これらの符号語には、データ列が含まれているので、これを含めるように復調部 1 1 1 で復調が行われる。

【0 1 4 1】同期信号識別部 2 2 1 は、同期信号のデータ部分を除くユニークパターンの部分のビットを指定する信号を同期信号除去部 2 2 2 に出カする。同期信号除去部 2 2 2 は、この信号に対応して同期信号ビット (ユニークパターンのビット) だけを除去する。

【0 1 4 2】図 1 1 に同期信号と DSV制御ビットを挿入した記録符号列の例を示す。この例では、同期信号として 2 4 符号語が用いられ、DSV制御は 5 6 データビット置きに行なわれ、5 回の DSV制御ごとに同期信号が挿入されている。このとき、同期信号ごとの符号語数 (チャネルビット数) は、
2 4 + (1 + 5 6 + 1 + 5 6 + 1 + 5 6 + 1 + 5 6 + 1 + 5 6 + 1) × 1. 5 = 4 5 3 符号語 (チャネルビット)

となる。このときのデータ語の冗長度は、次のように、約 7. 3 % となる。
(5 6 × 5) × 1. 5 / 4 5 3 = 4 2 0 / 4 5 3

$$= 0. 9 2 7$$

$$1 - 0. 9 2 7 = 0. 0 7 2 8$$

【0 1 4 3】発明者等は、以上の変換テーブルを用いた変調結果をシミュレーションしてみた。T min の連続を制限し、かつデータ列内において DSV制御ビットを挿入したデータ列を変調した結果について以下に示す。シミュレーションには、表 2 及び表 3 が用いられた。さらに比較のために、従来の RLL (1ー7) 変調である表 1 についてでもシミュレーションが行われた。

【0 1 4 4】任意に作成したランダムデータ 1 3 1 0 7 2 0 0 ビットを、5 6 データビットおきに DSV制御ビットを 1 ビットを挿入することで DSV制御した後、表 2 または表 3 の変換コードテーブルを用いて、符号語列 (チャネルビット列) に変換した場合の結果は以下の通りである。また同様に、任意に作成したランダムデータ 1 3 1 0 7 2 0 0 ビットを、表 1 の変換コードテーブルを用いて、符号語列 (チャネルビット列) に変換し、さらに 1 1 2 符号語 (チャネルビット) おきに、DSV制御ビットとして 2 チャネルビットを挿入することで DSV制御を行った時の結果は以下の通りである。

【0 1 4 5】ここで、表 2、表 3 では 5 6 データビットおき、また、表 1 では 1 1 2 符号語おきとしたのは、DSV制御ビットの冗長度を同一にするためである。このように、DSV制御の必要ビット数に差がある場合には、冗長度をそろえて考えた時、効率良く DSV制御が行える表 2 や表 3 の方が、表 1 に較べて低域特性が良好となる。
【0 1 4 6】また、各結果の数値は以下のよう

50

算した。
Ren_cnt[1 to 10]：最小ランの繰り返し 1 回乃至 1 0 回
の各発生数。
T_size[2 to 10]：2 T乃至 1 0 Tの各ランの発生数。
Sum：Number of bits. ビット総数。
Total：Number of runlengths. 各ラン (2 T, 3 T,
...) の発生総数
Average Run：(Sum/Total)
ラン分布の数値：(T_size[i] * (i)) / (Sum) , i=
2,3,4,,10
表 6 の 2 T乃至 1 0 Tの欄に示す数値が、このラン分布
の数値を表す。
最小ランの連続する分布の数値：(Ren_cnt[i] * (i))
/ T_size[2T],
i=1,2,3,4,,10
表 6 のRMTR(1)乃至RMTR(9)の欄に示す値が、この最小ラ *

<表 6 >

*** PPl7 comparison ***

	<表 2>		<表 3>	
	17PP-32		17PP-52	
Average Run	3.3665		3.4048	
Sum	20011947		20011947	
Total	5944349		5877654	
2T	0.2256		0.2246	
3T	0.2217		0.2069	
4T	0.1948		0.1935	
5T	0.1499		0.1491	
6T	0.1109		0.1094	
7T	0.0579		0.0814	
8T	0.0392		0.0351	
9T	-----		-----	
10T	-----		-----	
RMTR(1)	0.3837		0.3890	
RMTR(2)	0.3107		0.3137	
RMTR(3)	0.1738		0.1906	
RMTR(4)	0.0938		0.0806	
RMTR(5)	0.0299		0.0228	
RMTR(6)	0.0081		0.0033	
RMTR(7)	-----		-----	
RMTR(8)	-----		-----	
RMTR(9)	-----		-----	
max-RMTR	6		6	

peak DSV	# -36to36	# -35to40	* -46to43	* -1783to3433
	("#:56data-bit+1dc-bit, 1.75%)		(*:112cbit+2dc-cbit, 1.75%)	

*の連続する分布の数値を表す。
max-RMTR：最小ランの繰り返し返す、最大回数。
peak DSV：チャネルビット列のDSV制御を行う過程にお
いて、DSV値を計算したときのDSV値のプラス側のピーク
及びマイナス側のピークを言う。
DSV制御ビットとして 5 6 データ列おきにDSV制御ビット
を挿入した場合の冗長率は、5 6 データ列に対してDSV
制御ビット 1 ビットであるから、次のようになる。
1 / (1 + 5 6) = 1 . 7 5 %
また、DSV制御ビットとして 1 1 2 符号語列おきに 2 ビ
ットのDSV制御ビットを挿入した場合の冗長率は、1 1
2 符号語列に対してDSV制御ビット 2 ビットであるか
ら、次のようになる。
2 / (2 + 1 1 2) = 1 . 7 5 %
すなわち同じ冗長率である。
【 0 1 4 7 】

<表 1 >

+2bit-DC (Without-DCC)	
(DSV制御)	(DSV制御なし)
3.3016	3.2868
20011788	19660782
6061150	5981807
0.2417	0.2419
0.2234	0.2281
0.1902	0.1915
0.1502	0.1511
0.1135	0.1141
0.0561	0.0544
0.0218	0.0188
0.0023	-----
0.0009	-----
0.3628	0.3641
0.2884	0.2883
0.1717	0.1716
0.0909	0.0907
0.0456	0.0452
0.0219	0.0217
0.0100	0.0099
0.0047	0.0046
0.0022	0.0022
18	18

【 0 1 4 8 】 上の結果より、表 2 及び表 3 を用いると、
RL (1 , 7) 方式となっていること、最小ランと最大
ランが守られ、かつ最小ランの連続が最大で 6 回までに 50 k DSVの値が所定の範囲内に納められていること、こ
制限されていることが確認された。また、peak DSVの結果より、データ列内でDSV制御を行うことができる (pea

の場合、DSV制御ビットの効率がよいので、従来の符号語列（チャネルビット列）にDSV制御ビットを挟む方法よりも良好な低域成分を得ることができる（peak DSVの振れ幅が、表1の場合、89（＝46＋43）であるのに対して、表2の場合、72（＝36＋36）、表3の場合、75（＝35＋40）と、表1の場合の値より小さくなっている）ことが確認された。

【0149】以上より、従来のRLL（1ー7）方式（表1の方式）と比較すると、表2または表3を用いる方式（17PP方式）は、最小ランジの繰り返しが多くても6回までに制限することが出来るので、高線密度におけるエラー特性の改善を見込むことが出来る。

【0150】また、DSV制御の効率がよいので、同じ冗長度1.75%でDSV制御を行うと、17PP方式の方がピークDSV値の差が小さく出来、したがって、より低域成分を抑制することが可能となるので、安定したデータ記録再生を行うことが出来る。

＜表7＞

Shift error response

	＜表2＞ 17PP-32	＜表3＞ 17PP-52	＜表1＞ +2bit-DC
worst case (dc bit)	3 Bytes	3 Bytes	2 Bytes
Byte error(0)	in.	in.	without
Byte error(1)	0.028	0.096	0.080
Byte error(2)	0.777	0.635	0.826
Byte error(3)	0.195	0.268	0.094
Average -	0.000	0.001	-----
Byte error rate	1.167Byte	1.174Byte	1.014Byte

【0153】以上のように、この実施の形態は、最小ランジd＝1、最大ランジk＝7、変換率m／n＝2／3の変換テーブルにおいて、最小ランジ長の繰り返し回数を制限する置き換えコードを設けるようにしたので、

（1）高線密度での記録再生、及び、タンジェンシャル・チャルトに対する許容度が向上する。

（2）信号レベルが小さい部分が減少し、AGCやPLL等の波形処理の精度が向上し、総合特性を高めることができる。

（3）従来と比較して、ビタビ復号等の際のバスマemory長を短く設計することができ、回路規模を小さくすることが出来る。

【0154】また、変換テーブルの要素内の「1」の個数と、変換される符号語列の要素内の「1」の個数を、2で割った時の余りが、どちらも1あるいは0で一致するようにしたので、

（4）DSVの制御のための冗長ビットを少なくすることが出来る。

（5）最小ランジd＝1かつ(m,n)=(2,3)においては、1.5符号語でDSV制御を行うことが出来る。

（6）冗長度が少ないう上に、最小ランジと最大ランジを守る

* 【0151】さらに、シミュレーションにおいて、上述した場合と同一のランダムデータを使って発生させたチャネルビット列における、ビットシフト時の復調エラー伝搬特性を調べたところ、17PPの最悪エラー伝搬は、3バイトであるが、実際の発生頻度はほとんどないことが確認され、従来のRLL（1ー7）に較べての悪化はそれほどではない（平均バイトエラーレートは、表1の場合、1.014バイトであるのに対して、表2の場合、1.167バイトであり、表3の場合、1.174バイトである）ことが確認された。なお、このエラーレートの結果の数値は、本発明によるテーブルではDSV制御ビットを含み、また従来RLL（1ー7）では含んでいない。すなわち必ずしも同じ条件での測定とは言えず、これらが数値に影響を及ぼすことが考えられ、比較にあたってはその点を考慮する必要がある。

【0152】

*

＜表1＞
+2bit-DC
2 Bytes
without
0.080
0.826
0.094

1.014Byte

＜表2＞
17PP-32
3 Bytes
in.
0.028
0.777
0.195
0.000
1.167Byte

＜表3＞
17PP-52
3 Bytes
in.
0.096
0.635
0.268
0.001
1.174Byte

【0155】さらに本テーブルは特に、ランジ制限を守る置き換えコードを設けるようにしたので、

（7）テーブルがコンパクトになる。

（8）ビットシフト時の復調エラー伝搬を、表1の場合と殆ど同じ状態にすることが出来る。

【0156】なお、上記したような処理を行うコンパクトプログラムをユーザに提供する提供媒体としては、磁気ディスク、CD-ROM、固体メモリなどの記録媒体の他、ネットワーク、衛星などの通信媒体を利用することが出来る。

【0157】

【発明の効果】以上の如く、請求項1に記載の変調装置、請求項25に記載の変調方法、請求項26に記載の提供媒体、請求項27に記載の復調装置、請求項30に記載の復調方法、および請求項31に記載の提供媒体によれば、データ列の要素内の「1」の個数と、変換される符号語列の要素内の「1」の個数を、2で割った時の余りが、どちらも1あるいは0で一致するような変換規則、最小ランジdの連続を有限回以下に制限する第1の置き換えコード、およびランジ制限を守るための第2の置

き換えコードを有する変換テーブルで変換処理を行うようにしたので、少ない冗長度でDSV制御を行うことができるとともに、高線密度でエラーの少ない符号語列を記録再生することが可能となり、さらに、ビットシフト時の復調エラー伝搬の増加を抑制することができる。

【図面の簡単な説明】

【図 1】 本発明の変調装置の構成例を示すブロック図である。

【図 2】 図 1 の DSV 制御ビット決定・挿入部 1 1 の動作を説明する図である。

【図 3】 図 1 の変調部 1 2 の構成例を示すブロック図である。

【図 4】 図 3 の変調部 1 2 の動作を説明する図である。

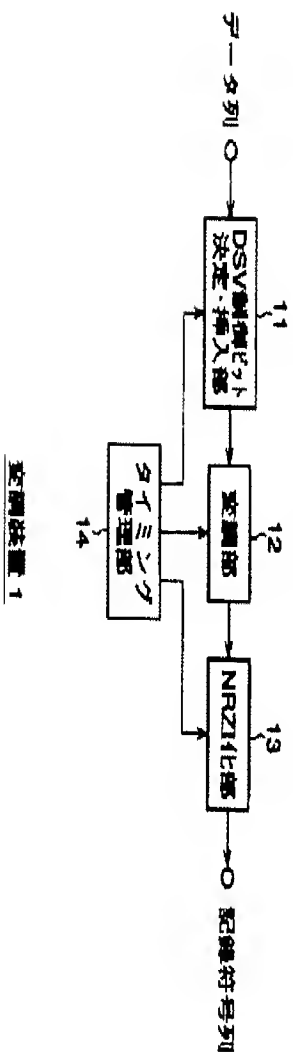
【図 5】 本発明の復調装置の構成例を示すブロック図である。

【図 6】 図 5 の復調部 1 1 1 の構成例を示すブロック図である。

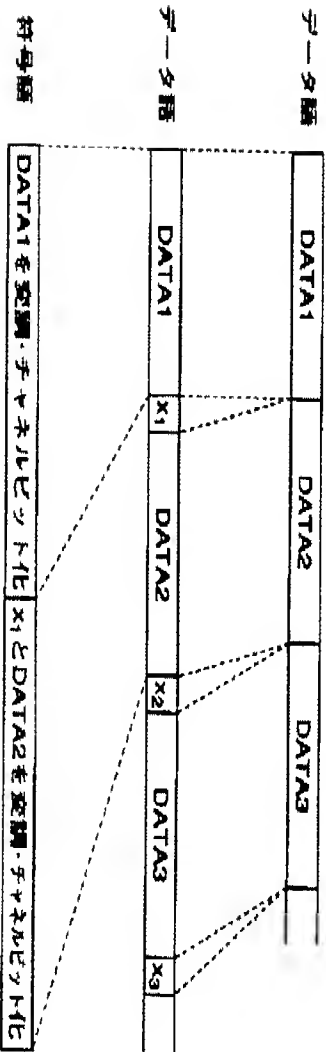
【図 7】 図 6 の復調部 1 1 1 の動作を説明する図である。

【図 8】 図 5 の DSV 制御ビット除去部 1 1 2 の処理を説明する図である。

【図 1】



【図 2】



【図 8】 図 5 の DSV 制御ビット除去部 1 1 2 の処理を説明するフローチャートである。

【図 9】 本発明の変調装置の他の構成例を示すブロック図である。

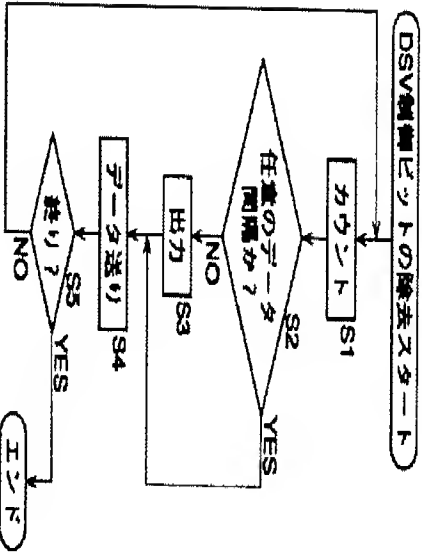
【図 10】 本発明の復調装置の他の構成例を示すブロック図である。

【図 11】 同期信号と DSV 制御ビットを挿入した記録符号列の例を示す図である。

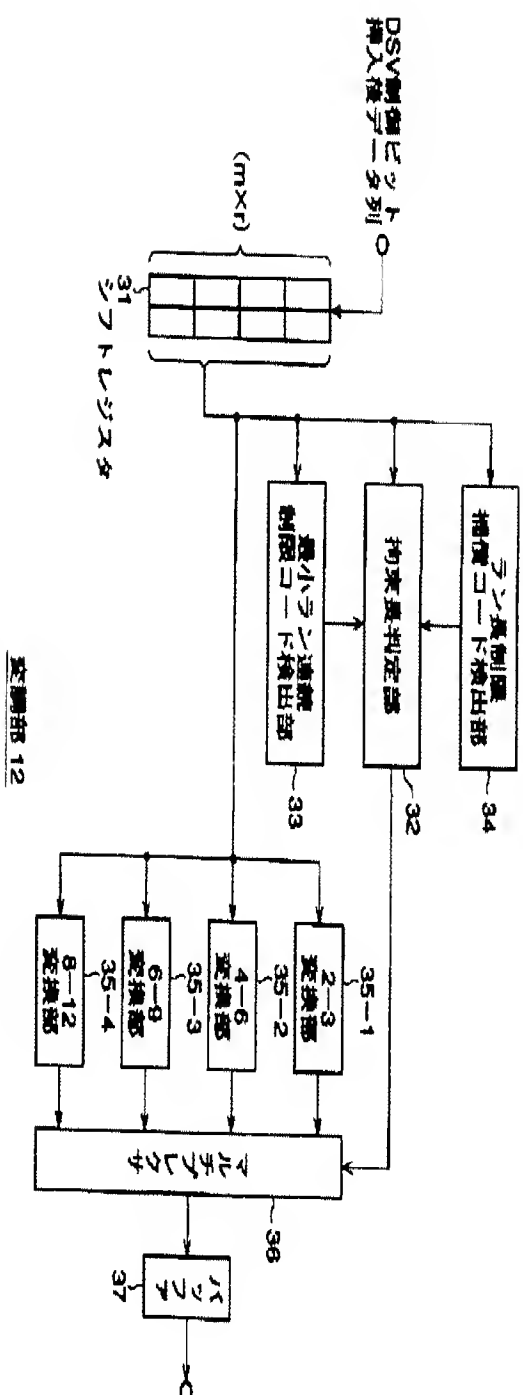
【符号の説明】

- 1 1 DSV 制御ビット決定・挿入部, 1 2 変調部, 1 3 NRZI 化部, 3 1 シフトレジスタ, 3 2 拘束長判定部, 3 3 最小ラン連続制限コード検出部, 3 4 ラン長制限補償コード検出部, 3 5 - 1 乃至 3 5 - 4 変換部, 3 6 マルチプレクサ, 3 7 バッファ, 1 1 1 復調部, 1 1 2 DSV 制御ビット除去部, 1 2 1 コンパレータ部, 1 2 2 拘束長判定部, 1 2 3 最小ラン連続制限コード検出部, 1 2 4 ラン長制限補償コード検出部, 1 2 5 - 1 乃至 1 2 5 - 4 逆変換部, 1 2 6 マルチプレクサ

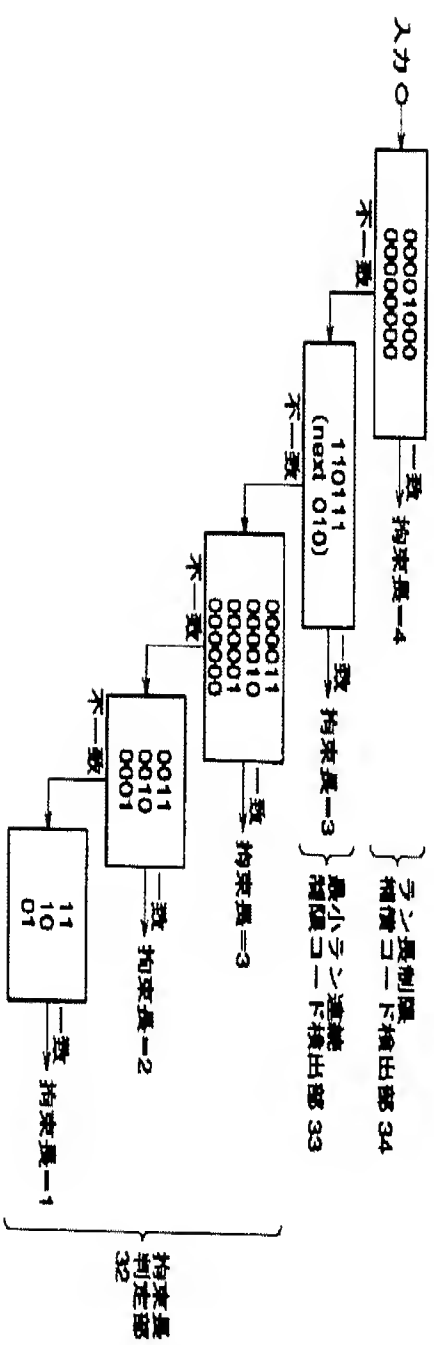
【図 8】



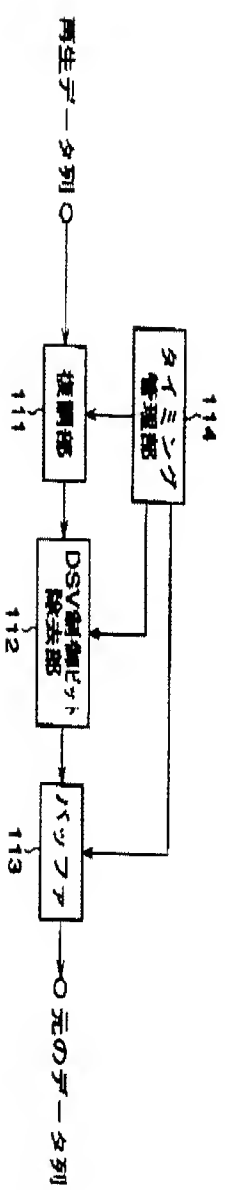
【例3】



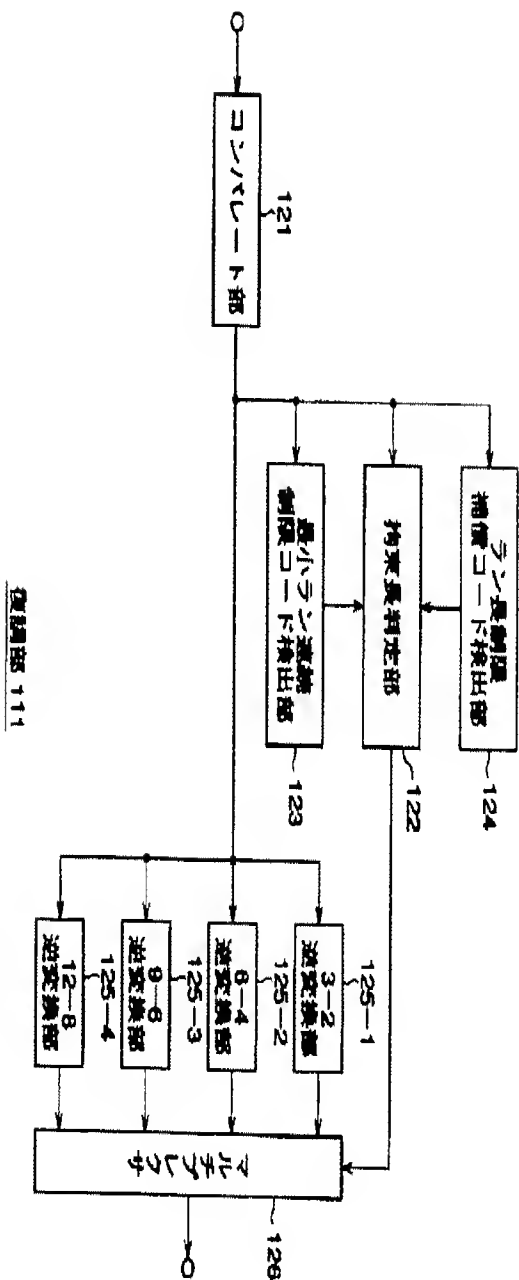
【例 4】



【5】

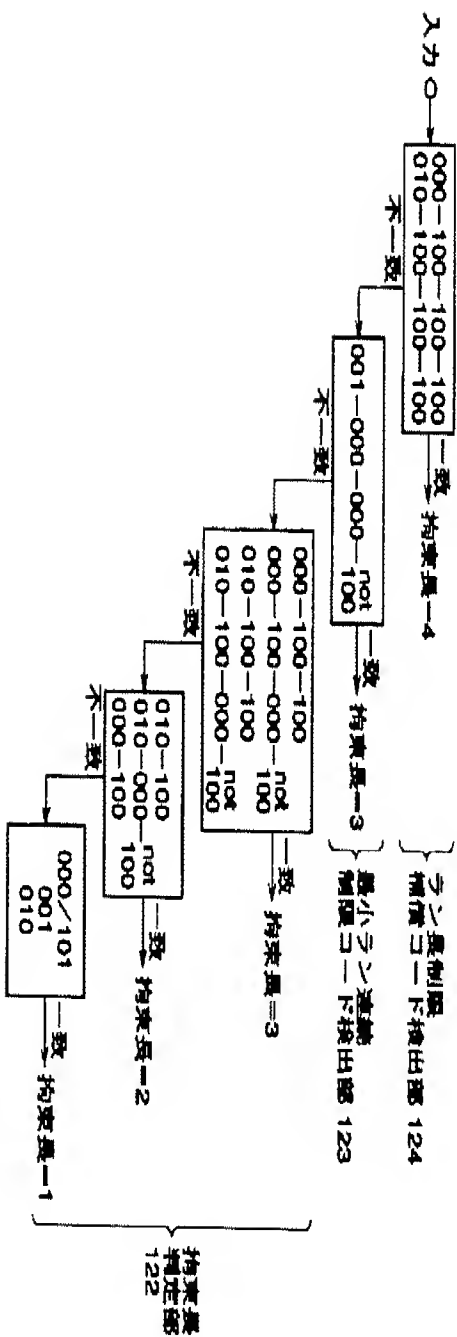


【図 6】

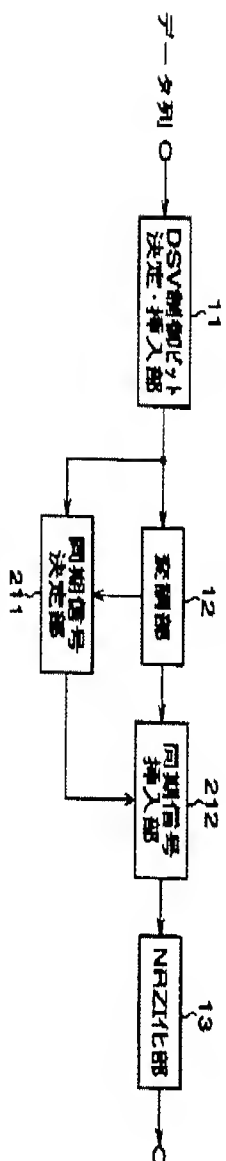


復調部 111

【図 7】



【図 9】



変調装置 1

